Evolutionary Learning

Changshui Zhang, Jianwen Zhang, Yangqing Jia

State Key Laboratory on Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology (TNList)
Department of Automation, Tsinghua University, Beijing 100084, China

November 7, 2009

Outline of Part I

Outline of Part I

Outline of Part I

1. A dilemma between learning algorithms and practical systems
2. Evolutionary learning
3. Analogs

Outline of Part II

Outline of Part II

Outline of Part II

4. Pioneer works
5. Evolutionary clustering
6. Online evolutionary exponential family mixture
   - A density estimation viewpoint to clustering
   - The roles of data and model: two general approaches
   - Experiments

Outline of Part III

Outline of Part III

Outline of Part III

Outline of Part III

Part I

Introduction

Static learning & dynamic systems

- Conventional learning on *static* data

$$\{X_{train}, X_{test}\} \overset{i.i.d.}{\sim} P(x)$$

- Practical systems: dynamic

$$\{X^1_{train}, X^1_{test}\} \overset{i.i.d.}{\sim} P_1(x)$$

$$\{X^2_{train}, X^2_{test}\} \overset{i.i.d.}{\sim} P_2(x)$$

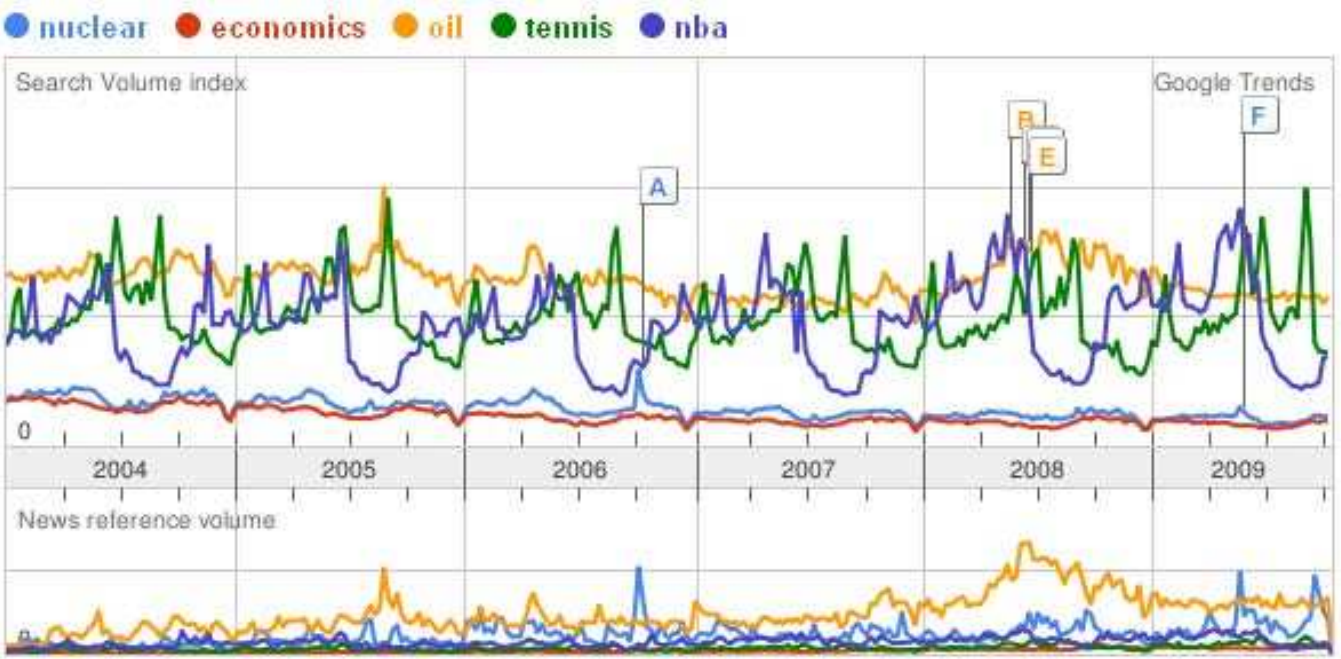
$$\ldots\ldots$$

$$\{X^t_{train}, X^t_{test}\} \overset{i.i.d.}{\sim} P_t(x)$$

$$P_1(x) \rightarrow P_2(x) \rightarrow \ldots \rightarrow P_t(x)$$

Examples

- Topics on Internet. *Google Trends*: five keywords (“nuclear”, “economics”, “oil”, “tennis”, “NBA”)



- Blog, social networks, vedio,...

- Dilemma
  "Static" is the typical setting we think about learning problems.
  "Dynamic" is the typical behavior of practical systems implementing learning algorithms.

Question:

Is it necessary to introduce a new algorithm especially for dynamic problems?

Possible solvers?

1. Applying static algorithms on the whole collection of data.
   - Non I.I.D.
2. Applying static algorithms on data at each time snapshot $X^t \rightarrow \mathcal{M}^t(x)$ or $p\left(\mathcal{M}^t|X^t\right)$.
   - Sample size (especially the labeled samples)
   - Time evolving mechanism.
   - Predicting (dynamic predicting rather than the classifier's predicting)

Evolutionary learning

Evolutionary learning

The learning problem on time evolving data.

Learning tasks

- Static performance

$$\sum_t loss\left(\mathcal{P}_t, \mathcal{M}_t\right)$$

- Learning time evolving mechanism

$$p\left(\mathcal{P}_{t+1} \middle| \mathcal{P}_t, \ldots, \mathcal{P}_1\right)$$

Taxonomy

- Online

$$\mathcal{M}_t | X_t, \ldots, X_1$$

- Off-line

$$\mathcal{M}_1, \ldots, \mathcal{M}_t, \ldots, \mathcal{M}_T \Big| X_1, \ldots, X_t, \ldots, X_T$$

- (Semi-)supervised: classification / regression
- Unsupervised: clustering / density estimation

Analogs

- Incremental learning
- Online learning
- Learning on time series

Incremental learning

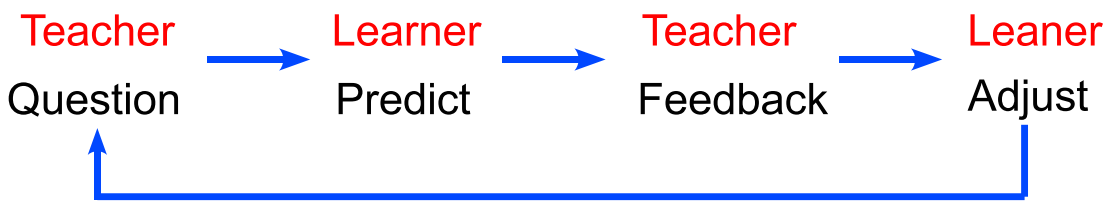
Data are organized into batch or stream mode due to limited memory or on-line setting, etc.

Differences from evolutionary learning

- Assumption: all data comes from an identical distribution.
- Output: a single output (classifier / regressor / data partition / density model /) for the entire data set.
- Not put forward for temporary data, on the contrary, a good incremental algorithm is expected to be insensitive to the order of data.

Online learning I

A special type of practical learning environment: a sequence of consecutive rounds. When predicting at t, leaner can only observe samples up to t. Leaner dynamically updates, to pursuit a low *regret*.



Difference between online learning and evolutionary learning.

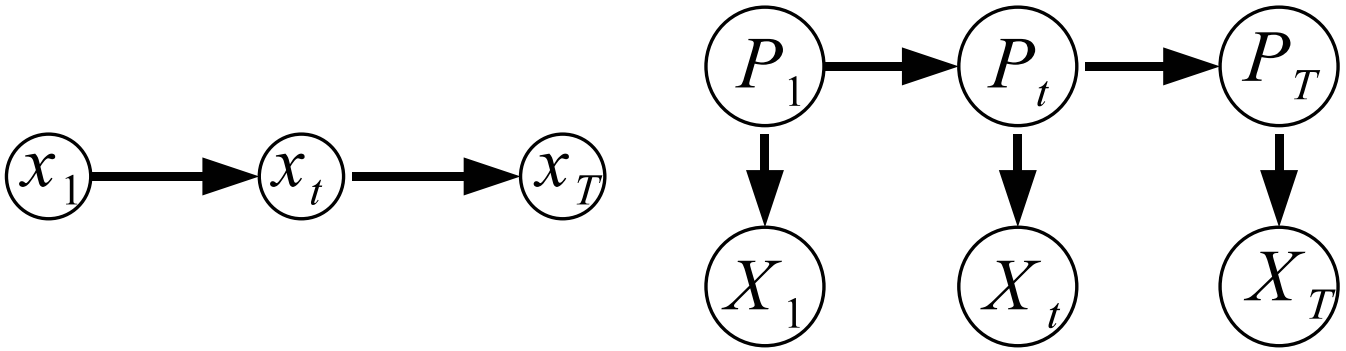
1. The data access of evolutionary learning can be online or off-line.

Online learning II

2. At each snapshot t, online learning observes a single sample, while evolutionary learning observe a package of samples.
3. Online learning requires a feedback for each sample, while in evolutionary learning, the package of samples may be labeled, or unlabeled, or partially labeled.
4. Online learning minimize a *regret* relative to the corresponding offline learning. No generalization problem exists at each snapshot t. Evolutionary learning aims to achieve two objectives: generalized loss at each snapshot t; the time evolving mechanism.

Learning on time series

- Time series: at t, a single observation x_t
- Evolutionary learning: at t, a distribution $\mathcal{P}_t$ (i.i.d. samples X_t)



Part II

Unsupervised Evolutionary Learning

Outline of Part II

Outline of Part II

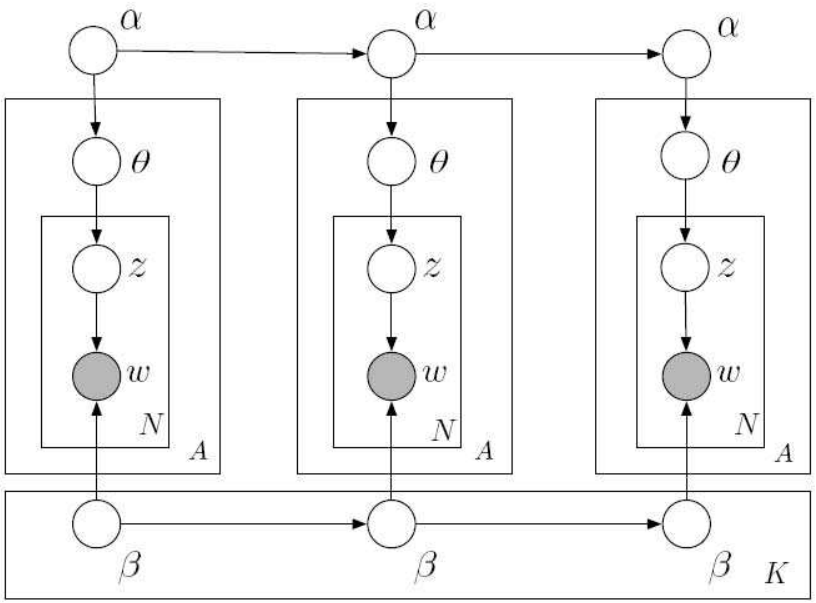
Outline of Part II

Pioneer works on evolutionary learning

- *Dynamic topic model* [BL06] (ICML'06)
- *Segmentation of image sequences* [GG06] (TPAMI'06)
- *Evolutionary clustering* [CKT06, CSZ+07, AX08] (KDD'06, KDD'07, SDM'08).

Dynamic topic model [BL06] (ICML'06)

A dynamic extension to the topic model *Latent Dirichlet Allocation* [BNJL03] (JMLR'03)



Segmentation of image sequence [GG06] (TPAMI'06)

Each frame image is modeled by a Bayesian mixture model.

Bayesian Gaussian mixture model

$$z \sim \mathit{Multnomial}(\alpha_1, \ldots, \alpha_K), \quad x \sim \mathcal{N}(\mu_z, \Sigma_z)$$
$$\alpha \sim \mathit{Dirichlet}(\gamma), \quad \left(\mu_z, \Sigma_z^{-1}\right) \sim \mathcal{NW}(\Theta_z)$$

where $\mathcal{NW}(\Theta_z)$ is the *Normal Wishart* distribution (the conjugate prior for Gaussian).

[GG06] let $\alpha_t \sim \mathit{Dirichlet}(\gamma_{t-1}), \quad \left(\mu_{z,t}, \Sigma_{z,t}^{-1}\right) \sim \mathcal{NW}\left(\Theta_{z,t-1}\right)$.
Notice: in both [BL06] and [GG06], component number is fixed along time.

Evolutionary Clustering

Arising from data mining [CKT06] (KDD'06).

The target of evolutionary clustering

- For each time epoch t, output a partition Π_t on X_t;
- A trade-off between good partition quality and preserving the smoothness of $\{\Pi_t\}_{t=1}^{T}$ along t.
- Cluster tracking.

Why is evolutionary clustering meaningful?

- Interpretability of machine learning and data mining algorithms.
- The importance of consistency between clustering results along time, especially for visualization.
- Clustering tracking provides powerful approach to dynamic network behavior analysis.

Why is evolutionary clustering challenging?

- The variation of data size.
- The variation of cluster number.
- Dynamic behavior of clusters along time, e.g., birth, merging, death, etc.

Evolutionary clustering of [CKT06]

An abstract framework

$$\max_{C_t} sq\left(C_t, M_t\right) - \lambda \cdot hc\left(C_{t-1}, C_t\right).$$

C_t: partition at t; M_t: similarity matrix between samples; sq: *snapshot quality*; hc: *historical cost*.

Evolutionary *k-means*

$$\min \sum_{i=1}^{n_t} \left\| x_i - m_{c(x_i)} \right\| + \lambda \sum_{c=1}^{C_t} \left\| m_c^t - m_{c'}^{t-1} \right\|$$

$$c' = \arg\min_z \left\| m_c^t - m_z^{t-1} \right\|$$

Evolutionary spectral clustering [CSZ+07]

- Original *Normalized Cut*

$$Cost_{NC} = k - \mathbf{Tr}\left[Z^\top \left(D^{-\frac{1}{2}} W D^{-\frac{1}{2}}\right) Z\right]$$

- Preserving Cluster Quality (PCQ)

$$\begin{aligned} Cost_{NC} &= \alpha \cdot NC_t\big|_{Z_t} + \beta \cdot NC_{t-1}\big|_{Z_t} \\ &= k - \mathbf{Tr}\left[Z_t^\top \left(\alpha D_t^{-\frac{1}{2}} W_t D_t^{-\frac{1}{2}} + \beta D_{t-1}^{-\frac{1}{2}} W_{t-1} D_{t-1}^{-\frac{1}{2}}\right) Z_t\right] \end{aligned}$$

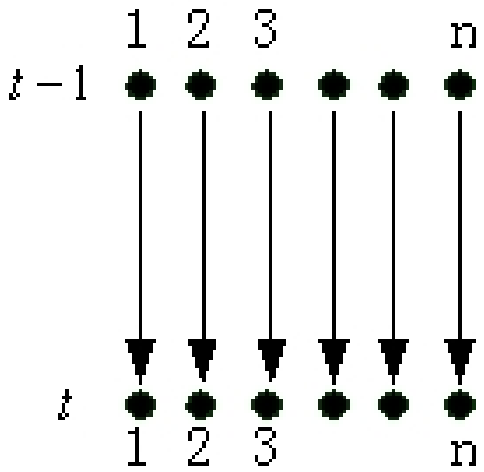
- Preserving Cluster Membership (PCM)

$$\begin{aligned} Cost_{NC} &= \alpha \cdot NC_t\big|_{Z_t} + \beta \cdot CT\left(Z_t, Z_{t-1}\right) \\ &= k - \mathbf{Tr}\left[Z_t^\top \left(\alpha D_t^{-\frac{1}{2}} W_t D_t^{-\frac{1}{2}} + \beta Z_{t-1} Z_{t-1}^\top\right) Z_t\right] \end{aligned}$$

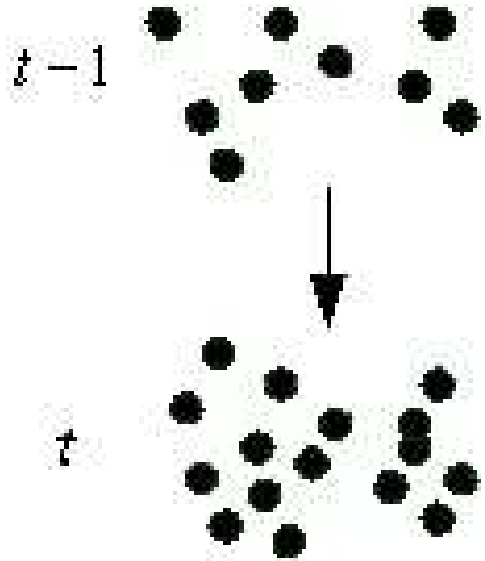
Two different settings about evolutionary clustering

[CSZ+07] differs with [CKT06] in the basic data setting:

- Evolutionary spectral clustering of [CSZ+07]



- Evolutionary clustering of [CKT06]



Problems not made clear

- What's evolving?
- What dose "smoothness" mean?
- Evolutionary k-means, evolutionary GMM, etc. Is there a general approach?
- The behavior of clusters
- The variation of cluster number and data size

Online evolutionary exponential family mixture(Jianwen Zhang, etc. IJCAI, 09)

- A density estimation viewpoint to clustering problem
- Online evolutionary exponential family mixture
- Experiments

Density estimation via mixture models

- Density estimation (Fisher-Wald setting)

$$\min_{\Xi} \mathcal{L}(\Xi) = -\int \log p(\mathbf{x};\Xi) dF(\mathbf{x})$$

$$= KL(f||p) - \int f(x)\log f(x)dx$$

- Mixture models

$$p(\mathbf{x};\Xi) = \sum_z^C \alpha_z p(\mathbf{x};\theta_z), \text{with} \sum_z \alpha_z = 1$$

Solver: Expectation-maximization (EM)

A varational upper bound

$$\mathcal{L}(\Xi) \leq \mathcal{G}\left(q_x(\cdot), \Xi\right)$$

where $q_x(\cdot)$ is a probability distribution for z.

$$\textbf{E}\text{-step: } q_{\mathbf{x}}^{[t+1]}(\cdot) \leftarrow \arg\min_{q_{\mathbf{x}}(\cdot)} \mathcal{G}(q_{\mathbf{x}}(\cdot), \Xi^{[t]})$$

$$\textbf{M}\text{-step: } \Xi^{[t+1]} \leftarrow \arg\min_{\Xi} \mathcal{E}(q_{\mathbf{x}}^{[t+1]}(\cdot), \Xi)$$

E-step: as a clustering problem

- If no additional constraints on $q_X(\cdot)$,

$$q_{\mathbf{x}}^{[t+1]}(z) = p(z|\mathbf{x}, \Xi^{[t]})$$

This case is called "soft-clustering";

- If we constraint that $q_X(z) \in \{0, 1\}$,

$$q_{\mathbf{x}}^{[t+1]}(z) = \mathbf{I}_{\left[z = \arg\max_z p(z|\mathbf{x};\Xi^{[t+1]})\right]}$$

This case is called "hard-clustering";

M-step

- Solution with closed form when using *exponential family mixture* (EFM):

$$\boldsymbol{\mu}_z^{[t+1]} = \nabla\Psi(\boldsymbol{\theta}_z^{[t+1]}) = \frac{\mathbf{E}_f\left[q_{\mathbf{x}}(z)T(\mathbf{x})\right]}{\mathbf{E}_f\left[q_{\mathbf{x}}(z)\right]}.$$

$$\alpha_z^{[t+1]} = \mathbf{E}_f\left[q_{\mathbf{x}}(z)\right]$$

- Exponential family mixture (EFM)

$$p(\mathbf{x};\Psi,\Xi) = \sum_z^C \alpha_z p_\Psi(\mathbf{x};\boldsymbol{\theta}_z)$$

$p_\Psi(\mathbf{x};\boldsymbol{\theta}_z)$ belongs to a same *expoenntial family*:

$$p_\Psi(\mathbf{x};\boldsymbol{\theta}) = \exp\left\{\langle\boldsymbol{\theta}, T(\mathbf{x})\rangle - \Psi(\boldsymbol{\theta})\right\} p_0(T(\mathbf{x})).$$

Gaussian, Multinomial, Poisson, Binomial, Exponential, Dirichlet, ...

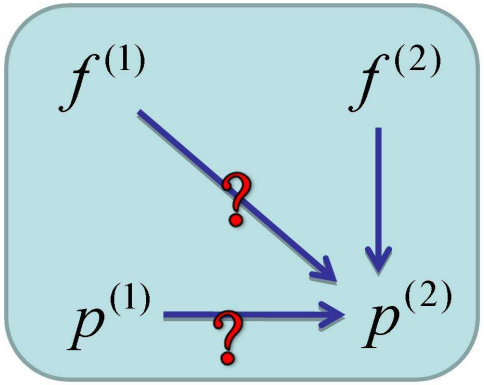
Two general approaches

- Data & model:
  Static: $f(x), p(x)$
  Evolutionary: $f^{(1)}(x), f^{(2)}(x), p^{(1)}(x), p^{(2)}(x)$
- Two possible approaches

$$f(x) \longrightarrow p(x)$$

$$\min KL(f \| p)$$

$$\mathcal{L} = (1-\lambda)\cdot dist\left(f^{(2)}, p^{(2)}\right) + \lambda \cdot dist\left(f^{(1)}, p^{(2)}\right) \quad \lambda \cdot dist\left(p^{(1)}, p^{(2)}\right)$$



Two possible approaches

Historical data dependent (HDD)
Historical model dependent (HMD)

Historical data dependent (HDD)

- Loss function

$$\begin{aligned}\mathcal{L}_d &= (1-\lambda)\cdot KL(f^{(2)}||p^{(2)}) + \lambda \cdot KL(f^{(1)}||p^{(2)}) \\ &= -\mathbf{E}_{\tilde{f}_\lambda}[\log p^{(2)}(\mathbf{x};\Xi^{(2)})]\end{aligned}$$

  where $\tilde{f}_\lambda(x) = (1-\lambda)f^{(2)}(\mathbf{x}) + \lambda f^{(1)}(\mathbf{x})$.

- Meaning
  Using EFM $p^{(2)}$ to estimate $\tilde{f}_\lambda$. (Recall the static clustering as using EFM p to estimate f: $\mathcal{L} = -\mathbf{E}_f[\log p(\mathbf{x};\Xi)]$)

- Solver
  EM.

Solution to HDD

- E-step: the same as clustering using EFM.
- M-step

$$\alpha_z^{(2),[t+1]} = \mathbf{E}_{\tilde{f}_\lambda}\left[q_{\mathbf{x}}^{[t+1]}(z)\right]$$

$$\boldsymbol{\mu}_z^{(2),[t+1]} = \frac{\mathbf{E}_{\tilde{f}_\lambda}\left[q_{\mathbf{x}}^{[t+1]}(z)T(\mathbf{x})\right]}{\mathbf{E}_{\tilde{f}_\lambda}\left[q_{\mathbf{x}}^{[t+1]}(z)\right]}$$

Historical model dependent (HMD)

- Loss function

$$\mathcal{L}_m = (1-\lambda)KL(f^{(2)}, p^{(2)}) + \lambda d_{\text{EMD}}(p^{(1)}, p^{(2)}).$$

where

$$d_{\text{EMD}}(p^{(1)}, p^{(2)}) = \min_{\mathbf{w}} \sum\nolimits_{l,z} w_{lz} d(p(\mathbf{x}; \theta_l^{(1)}), p(\mathbf{x}; \theta_z^{(2)}))$$

$$s.t. \quad w_{lz} \geq 0, \sum\nolimits_z w_{lz} = \alpha_l^{(1)}, \sum\nolimits_l w_{lz} = \alpha_z^{(2)}$$

The equivalent problem:

$$\min_{\Xi^{(2)}, \mathbf{w}} \mathcal{L}'_m(\Xi^{(2)}, \mathbf{w}) = (1-\lambda)KL(f^{(2)}(x), p^{(2)}(x; \Xi^{(2)})) + \lambda \sum\nolimits_{l,z} w_{lz} KL(p(x; \theta_l^{(1)}), p(x; \theta_z^{(2)}))$$

$$s.t. \ldots.$$

Optimization

- Variational upper bound by introducing $q_x(z)$:

$$\mathcal{L}'_m(\mathbf{w}, \Xi^{(2)}) \leq \mathcal{G}\left(q_{\mathbf{x}}(\cdot), \Xi^{(2)}, \mathbf{w}\right)$$

- Alternatively optimize w.r.t. $q_x(z)$, $\Xi^{(2)}$ and $\mathbf{w}$.
- q-step: posterior
- $\mathbf{w}$-step: $\min_{\mathbf{w}} \sum_{l,z} w_{lz} KL\left(p(\mathbf{x}; \theta_l^{(1)}), p(\mathbf{x}; \theta_z^{(2)})\right)$ s.t. ...
- Ξ-step:

$$\alpha_z^{(2),[t+1]} = \mathbf{E}_{f^{(2)}}[q_{\mathbf{x}}^{[t+1]}(z)]$$

$$\boldsymbol{\mu}_z^{(2),[t+1]} = \frac{(1-\lambda)\mathbf{E}_{f^{(2)}}[q_{\mathbf{x}}^{[t+1]}(z)T(\mathbf{x})] + \lambda \sum_l w_{lz}^{[t+1]} \boldsymbol{\mu}_l^{(1)}}{(1-\lambda)\mathbf{E}_{f^{(2)}}[q_{\mathbf{x}}^{[t+1]}(z)] + \lambda \sum_l w_{lz}^{[t+1]}}$$

- Data: NSF research awards abstracts.
  - 13 years, D = 19728, W = 15412
  - Features
    - Word count (For Mixture of multinomial)
    - *tf-idf* (For *k-means*)
- Models
  - *k-means* $\rightarrow$ Evolutionary k-means
  - Multinomial mixture model $\rightarrow$ Evolutionary MMM

Evaluation

- Normalized mutual information (NMI) at each epoch

$$NMI = \frac{I(Y;\hat{Y})}{\sqrt{H(Y)\cdot H(\hat{Y})}} \approx \frac{\sum_{h,c} \log\left(\frac{n\cdot n_{h,c}}{n_h n_c}\right)}{\sqrt{\left(\sum_h n_h \log\frac{n_h}{n}\right)\left(\sum_c n_c \log\frac{n_c}{n}\right)}}$$

- Data measured temporal loss (DTL)

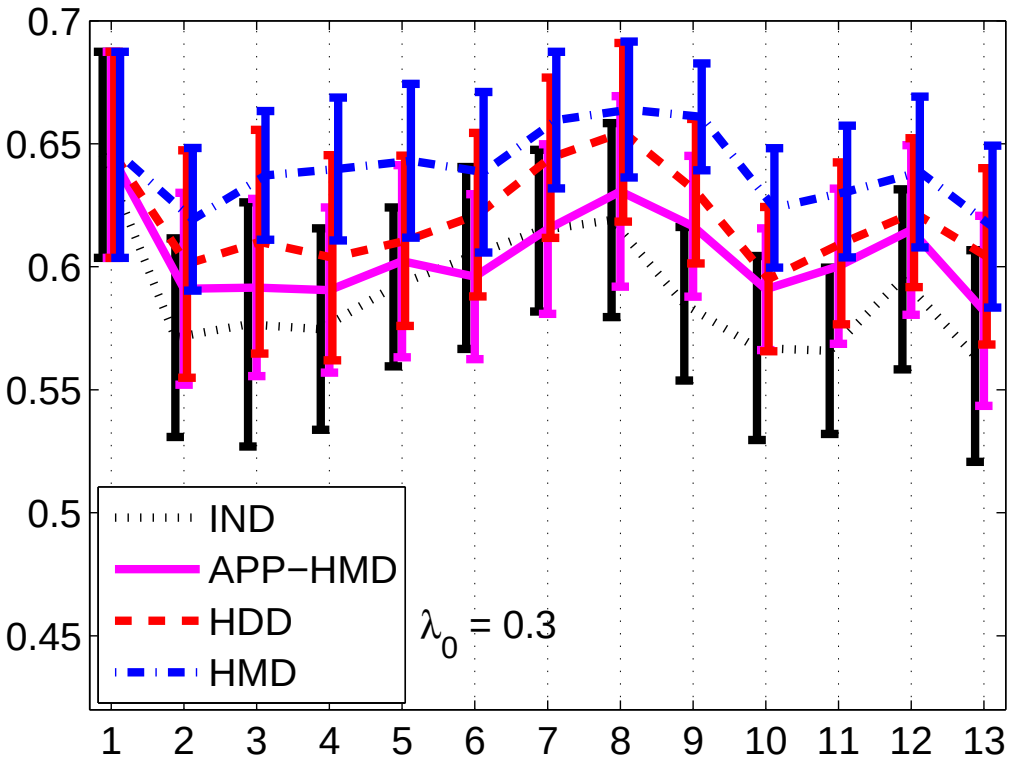
$$DTL = -\mathbf{E}_{f^{(1)}}\left[\log p^{(2)}(\mathbf{x})\right] = KL\left(f^{(1)}||p^{(2)}\right) - Const$$

- Model measured temporal loss (MTL)

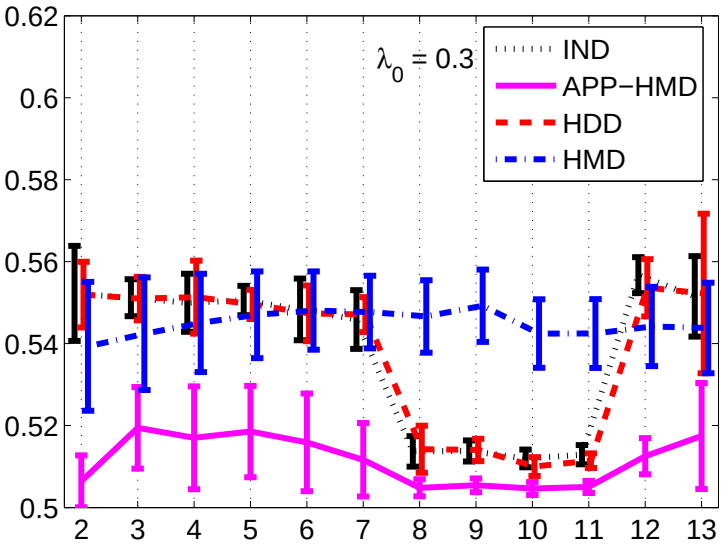
$$MTL = d_{\mathrm{EMD}}\left(p^{(1)}, p^{(2)}\right)$$

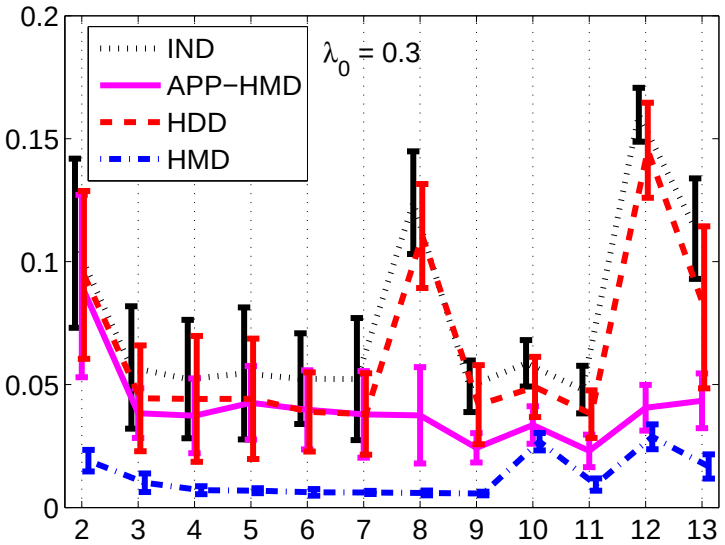
Results: Evolutionary k-means

- Snapshot performance: NMI



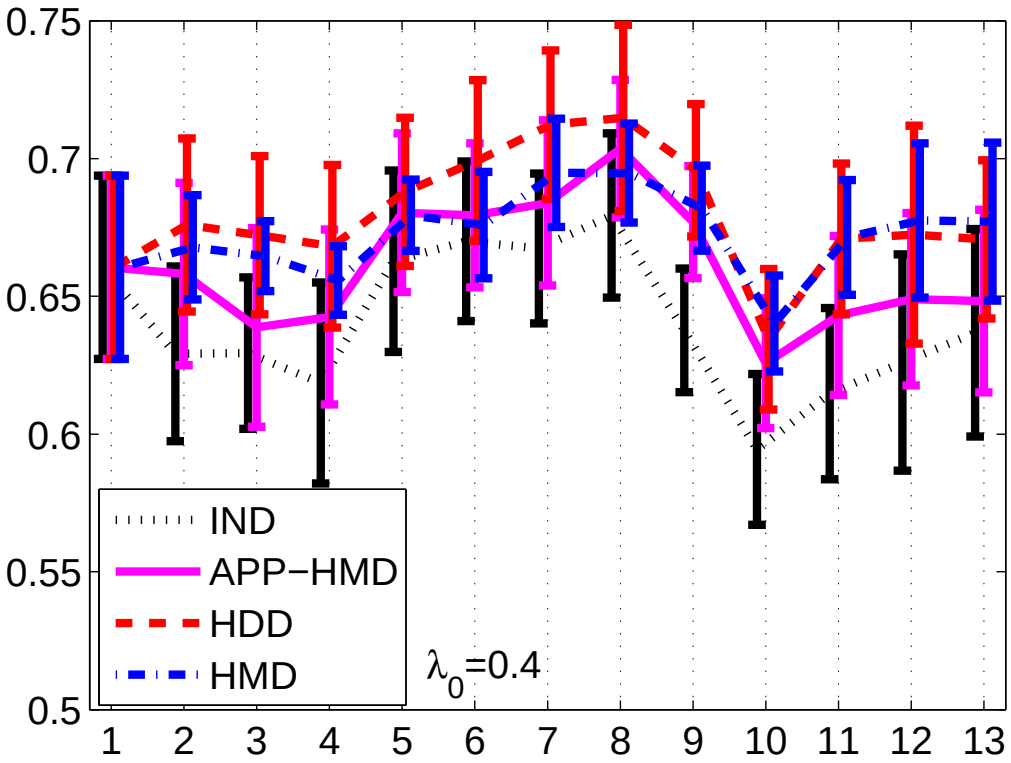
- Temporal performance: DTL & MTL



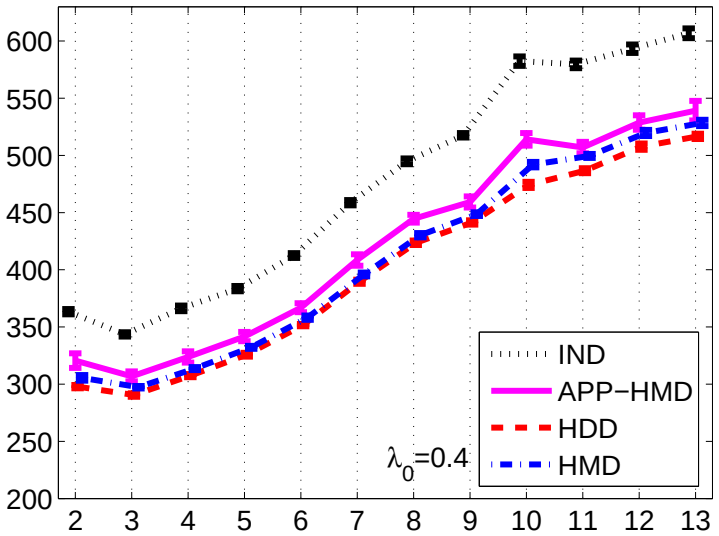


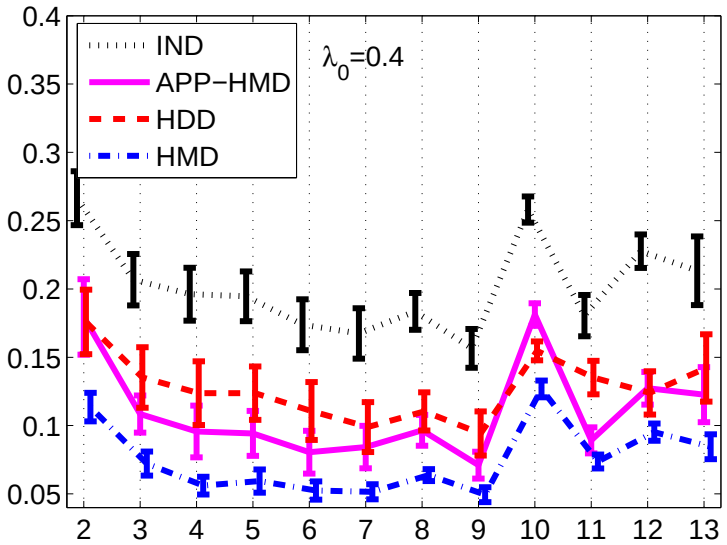
Results: Evolutionary MMM

- Snapshot performance: NMI



- Temporal performance: DTL & MTL





Summary

- A density estimation view point to clustering and evolutionary clustering problem
- Two general approaches based on exponential family mixture models
- Validated by different EFMs on real text data
- Unsolved problem: tracking of clusters

Part III

Supervised Evolutionary Learning

Outline of Part III

Outline of Part III

Outline of Part III

Outline of Part III

Evolutionary classification

Evolutionary classification

Evolutionary classification learns a chain of evolving classifiers for different time periods.

Why necessary?

- Using historical classifiers makes little sense:
  $\longrightarrow$ i.i.d. assumption dose not hold.
- Using all historic data also makes little sense:
  $\longrightarrow$ i.i.d. assumption dose not hold.
- Historical classification information may help:
  $\longrightarrow$ distribution evolves slowly (assumed but make sense)
- Historical labels may be utilized rather than discarded

A Brief Review of SSL

- Data: $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_l, \mathbf{x}_{l+1} \cdots, \mathbf{x}_n\}$, $\mathcal{Y} = \{y_i\}_{i=1}^{l}$.
- Require: $\{y_i\}_{i=l+1}^{n}$, or $f(\mathbf{x})$.
- Representative Work: Manifold Regularization [BNS06]:

$$\min_f \mathcal{J}(f) = \sum_{i=1}^{l} \mathcal{L}(y_i, \mathbf{x}_i, f) + \gamma_A \|f\|_K^2 + \gamma_I \|f\|_I^2 \qquad (1)$$

where

$$\|f\|_I^2 = \frac{1}{n^2} \sum_{i,j=1}^{n} (f(\mathbf{x}_i) - f(\mathbf{x}_j))^2 W_{ij} = \frac{1}{n^2} \mathbf{f}^\top \mathbf{L} \mathbf{f} \qquad (2)$$

is the spatial regularizer calculated via graph Laplacian.

Semi-supervised Evolutionary Classification: Basic Settings(Yangqing Jia. etc. IJCAI, 09)

- In general, we aim to learn a function $\mathcal{F}(t) : \mathbb{R} \mapsto \mathcal{H}_K$, where $\mathcal{H}_K$ is the Hilbert space of functions $f : \mathbb{R}^d \mapsto \mathbb{R}$ associated with a pre-defined kernel K.
- Specifically, $\mathcal{F}(t) = f_t$ gives the classification function for each time t.
- Data: $\mathcal{X} = \{\mathcal{X}_1, \mathcal{X}_2, \cdots, \mathcal{X}_T\}$ from T discrete "frames", where each $\mathcal{X}_t = \{\mathbf{x}_{1,t}, \mathbf{x}_{2,t}, \cdots, \mathbf{x}_{n_t,t}\}$, with the first l_t labeled as $\mathcal{Y}_t = \{y_i\}_{i=1}^{l_t}$.
- When the time t takes discrete values, the goal is to find T classification functions $\{\mathcal{F}(t)\}_{t=1}^{T} = \{f_t(\mathbf{x})\}_{t=1}^{T}$.

Evolutionary Smoothness Assumption

Similar to the smoothness assumption in the general learning problems, we extend it to the evolutionary data.

Smoothness Assumption

Two data points are likely to have similar labels if they are close to each other.

Evolutionary Smoothness Assumption

Two classification functions f_{t_1} and f_{t_2} are likely to be similar if the times t_1 and t_2 are close.

Temporal Regularizer

- A direct approach to carry out the assumption above is to use the integral of $\|\partial \mathcal{F}/\partial t\|^2$ over time t:

$$\int_1^T \left\| \frac{\partial \mathcal{F}}{\partial t} \right\|^2 \mathrm{d}t \tag{3}$$

- When t is discrete, we use the backward difference to approximate the above integral as

$$\int_1^T \left\| \frac{\partial \mathcal{F}}{\partial t} \right\|^2 \mathrm{d}t \approx \frac{1}{T-1} \sum_{t=2}^{T} \|f_t(\cdot) - f_{t-1}(\cdot)\|_K^2 \tag{4}$$

- This serves as a temporal regularizer for the learning algorithm.

Offline Algorithm

- Taking into consideration the evolutionary smoothness assumption, we propose an offline learning algorithm simultaneously learns the T classification functions by minimizing the following objective function w.r.t. $\{f_t\}_{t=1}^{T}$:

$$\mathcal{J}_{\text{off}} = \sum_{t=1}^{T} \left[\sum_{i=1}^{n_t} \mathcal{L}(y_{i,t}, \mathbf{x}_{i,t}, f_t(\cdot)) + \gamma_I \|f_t(\cdot)\|_I^2 \right] + \gamma_A \|f_1\|_K^2 + \gamma_A \sum_{t=2}^{T} \|f_t(\cdot) - f_{t-1}(\cdot)\|_K^2 \quad (5)$$

- It is not difficult to observe that the problem is convex if the loss function $\mathcal{L}$ is convex, thus the global optimal solution can be found.

Online Algorithm

- As the data accumulates, the learning problem scales up fast and renders the offline algorithm impractical.
- Thus, we propose an online algorithm that learns one classifier for the current frame with the historic information fixed. For time t, the algorithm minimizes the following objective function w.r.t. f_t:

$$\begin{aligned}\mathcal{J}_{\text{on}}^{(t)} &= \sum_{i=1}^{n_t} \mathcal{L}(y_{i,t}, \mathbf{x}_{i,t}, f_t(\cdot)) + \gamma_I \|f_t(\cdot)\|_I^2 \\ &\quad + \gamma_A \|f_t(\cdot) - f_{t-1}(\cdot)\|_K^2 \end{aligned} \tag{6}$$

The Representer Theorem

It can be proved that in the evolutionary case, the representer theorem still holds, enabling us to convert the problem from finding the functions $\{f_t\}_{t=1}^{T}$ to simply finding a set of expansion coefficients.

Representer Theorem

Assume that the regularizers $\|f_t(\cdot)\|_I^2$ $(1 \leq t \leq T)$ are carried out empirically using the graph Laplacian as $\mathbf{f}_t^\top \mathbf{L}_t \mathbf{f}_t$, then for each frame t, the minimizer of (6) admits an expansion

$$f_t^*(\mathbf{x}) = \sum_{k=1}^{t} \sum_{i=1}^{n_t} \alpha_{i,k} K(\mathbf{x}, \mathbf{x}_{i,k}), \quad \forall 1 \leq t \leq T \tag{7}$$

given a predefined Mercer kernel $K(\cdot, \cdot)$.

Solution Space Shrinking

- Problem: as the data accumulates, the scale of the expansion subspace $\mathcal{S} \subset \mathcal{H}_K$ that is constructed by the kernel functions $\{K(\cdot, \mathbf{x}_{i,k})\}_{i=1}^{n_k}{}_{k=1}^{t}$ may grow too large for learning.
- Thus, we manually impose a *representer constraint* to the optimization problem by shrinking the solution space:

$$f^*(\cdot; t) = \arg\min_{f_t(\cdot) \in \mathcal{S}_t} \mathcal{J}_{\text{on}}^{(t)} \tag{8}$$

where $\mathcal{S}_t = \text{span}\{K(\cdot, x_{i,t}) | x_{i,t} \in \mathcal{X}_t\}$ is spanned by the kernel functions corresponding to the dataset $\mathcal{X}_t$

Kernel Based Representation

- $f_t(\cdot) = \sum_{i=1}^{n_t} \alpha_{i,t} K(\cdot, \mathbf{x}_{i,t})$
- The objective function can be written as

$$\begin{aligned} \mathcal{J}_{\text{on}}^{(t)} &= (\mathbf{K}_t\boldsymbol{\alpha}_t - \mathbf{y}_t)^\top \mathbf{C}_t(\mathbf{K}_t\boldsymbol{\alpha}_t - \mathbf{y}_t) + \gamma_A \boldsymbol{\alpha}_t^\top \mathbf{K}_t \boldsymbol{\alpha}_t \\ &\quad - 2\gamma_A \boldsymbol{\alpha}_t^\top \mathbf{K}_{t,t-1}\boldsymbol{\alpha}_{t-1} + \frac{\gamma_I}{n_t^2}\boldsymbol{\alpha}_t^\top \mathbf{K}_t^\top \mathbf{L}_t \mathbf{K}_t \boldsymbol{\alpha_t} \end{aligned} \quad (9)$$

- And the solution is

$$\begin{aligned} \boldsymbol{\alpha}_t^* &= (\mathbf{K}_t^\top(\mathbf{C}_t + \frac{\gamma_I}{n_t^2}\mathbf{L}_t)\mathbf{K}_t + \gamma_A \mathbf{K}_t)^{-1} \\ &\quad (\mathbf{K}_t\mathbf{C}_t\mathbf{y}_t + \gamma_A \mathbf{K}_{t,t-1}\boldsymbol{\alpha}_{t-1}) \end{aligned} \quad (10)$$

Toy Data

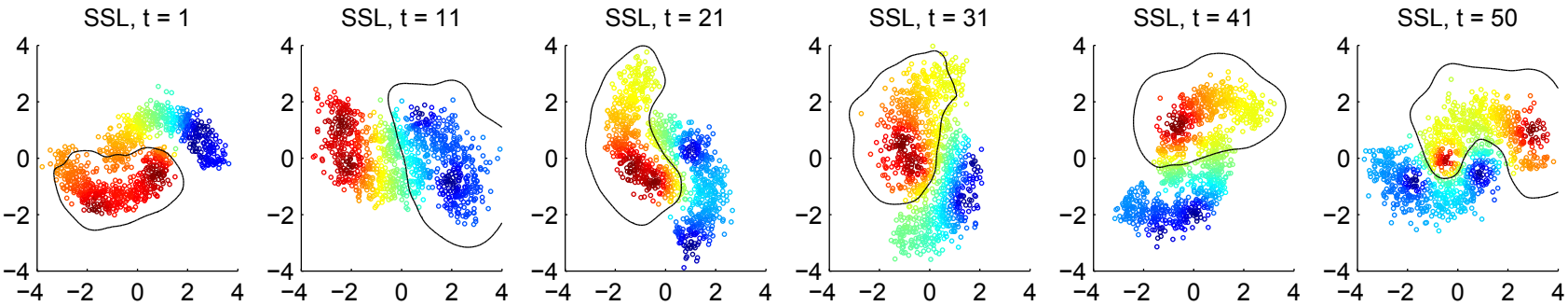


Figure: Standard SSL on each frame

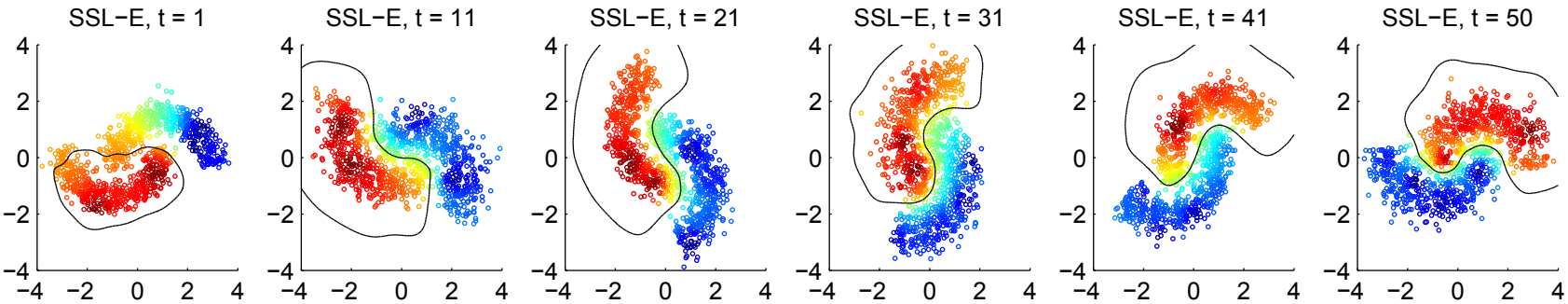


Figure: SSL-E

Real-world Dataset

- We crawled posts from 5 online mailing lists dated from 2003.9 to 2008.9, each month as a time frame.
- There are 161,675 posts (data points) for the learning task.
- We applied two other methods to compare with our algorithm:
  - classical SSL algorithm performed on each frame;
  - a naive evolutionary SSL extension that simply uses the data from the current frame and its preceding frame.

Label	Date	Total Posts	Posts/Month
audiophiles	2005.6 - 2008.9	35,666	892
listening-l	2003.7 - 2008.9	25,738	422
sqlite-users	2003.10 - 2008.9	32,655	544
tutor-python	2004.12 - 2008.9	26,314	572
wine-devel	2003.9 - 2008.9	41,302	677

Experimental Results: Statistics

Table: Experimental results on the evolutionary mailing list data using SSL, SSL-naive, and SSL-E.

class	aud/lis	aud/sql	aud/tut	aud/win
SSL	97.03 (4.42)	96.12 (5.19)	94.04 (7.05)	93.14 (8.20)
SSL-naive	97.77 (2.98)	95.99 (6.39)	95.47 (6.33)	93.18 (10.28)
SSL-E	**98.98** (2.03)	**98.80** (2.97)	97.95 (4.87)	**97.11** (6.89)
class	lis/sql	lis/tut	lis/win	sql/tut
SSL	84.40 (9.02)	81.50 (9.89)	78.41 (10.52)	63.00 (5.26)
SSL-naive	86.30 (7.39)	85.77 (8.38)	81.19 (9.86)	64.52 (3.41)
SSL-E	**92.74** (6.41)	**91.30** (6.65)	**91.41** (6.89)	**76.87** (4.10)
class	sql/win	tut/win	Multi-class	
SSL	64.42 (4.97)	63.33 (4.40)	77.49 (5.02)	
SSL-naive	68.45 (4.16)	65.57 (3.51)	78.13 (4.89)	
SSL-E	**80.87** (8.16)	**78.56** (7.13)	**86.10** (3.21)	

Experimental Results: Spatial and Temporal Loss

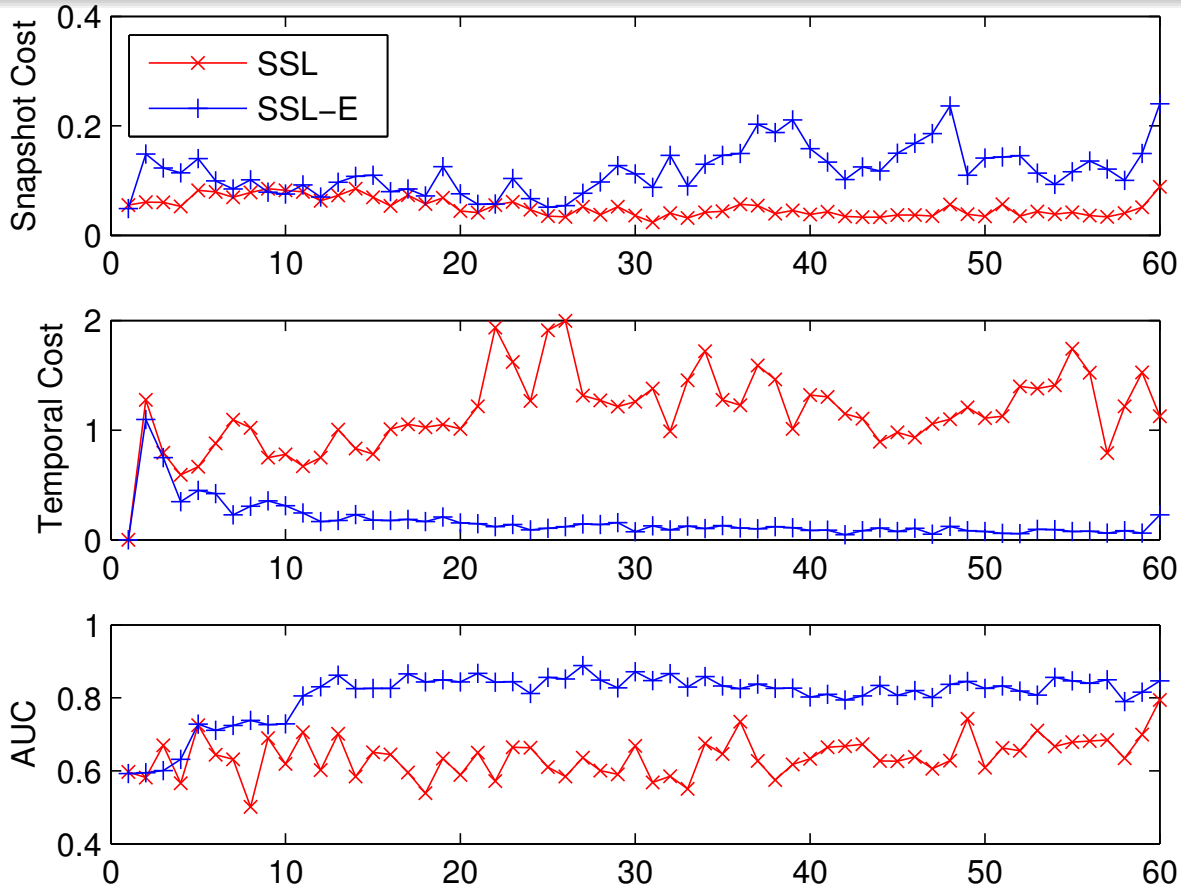


Figure: The spatial cost, temporal cost and AUC value vs. time on the *sql/win* data.

Summary

- Learning the natural evolutionary information of the data is a new challenge in the machine learning research:
  - the concept drifts during the time and makes a single aggregated classifier inaccurate for long-term prediction;
  - the drifting is smooth in a short time period.
- We proposed a new semi-supervised algorithm for learning a series of evolving classification functions for evolutionary data.
- The proposed algorithm provides much better performances on real-world application in both stability and accuracy.

Part IV

Conclusion

Conclusion

- A new type of learning problems goes beyond the I.I.D. assumption. We call it *Evolutionary Learning*.
- A semi-supervised classification algorithm.
- An unsupervised algorithm for clustering / density estimation.
- Just exploratory works. No theoretical analysis provided.

References I

Amr Ahmed and Eric Xing.

Dynamic non-parametric mixture models and the recurrent Chinese restaurant process: with applications to evolutionary clustering.
SIAM Conference on Data Minning, 2008.

D.M. Blei and J.D. Lafferty.

Dynamic topic models.
23rd International Conference on Machine Learning (ICML), 2006.

D.M. Blei, A.Y. Ng, M.I. Jordan, and J. Lafferty.

Latent Dirichlet allocation.
Journal of Machine Learning Research, 3(4-5):993–1022, 2003.

M. Belkin, P. Niyogi, and V. Sindhwani.

Manifold regularization: A geometric framework for learning from labeled and unlabeled examples.
Journal of Machine Learning Research, 7:2434, 2006.

Deepayan Chakrabarti, Ravi Kumar, and Andrew Tomkins.

Evolutionary clustering.
In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pages 554–560. ACM Press New York, NY, USA, 2006.

References II

Yun Chi, Xiaodan Song, Dengyong Zhou, Koji Hino, and Belle L. Tseng.

Evolutionary spectral clustering by incorporating temporal smoothness.
In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*. ACM Press New York, NY, USA, 2007.

J. Goldberger and H. Greenspan.

Context-based segmentation of image sequences.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(3):463–468, 2006.